

IN THE CLAIMS

1. (original) A method of equalizing a signal comprising at least the following steps:

a) shifting data in a series of input data blocks to the left;

b) complex multiplying each of the left shifted data blocks by a first set of equalizer coefficients to provide first adjusted output data blocks, wherein step b) is not a full solution to ghosts;

c) complex multiplying each of the input data blocks by a second set of equalizer coefficients to provide second adjusted output data blocks, wherein step c) is not a full solution to ghosts;

d) shifting the data in each of the input data blocks of data to the right;

e) complex multiplying each of the right shifted input data blocks by a third set of equalizer coefficients to provide third adjusted output data blocks, wherein step e) is not a full solution to ghosts; and,

f) adding corresponding ones of the first, second, and third adjusted output data blocks so as to provide a substantially full solution to ghosts.

2. (original) The method of claim 1 wherein step a) comprises the step of down sampling each of the input data blocks, wherein step c) comprises the step of down sampling each of the input data blocks, and wherein step d) comprises the step of down sampling each of the input data blocks.

3. (original) The method of claim 1 further comprising the step of g) applying a spectral transformation prior to steps a), c), and d), wherein the spectral transformation is longer than an input data block.

4. (original) The method of claim 1 further comprising the step of g) applying pre-processing coefficients to each data block prior to steps a), c), and d).

5. (original) The method of claim 4 wherein the pre-processing coefficients are curved.

6. (original) The method of claim 4 wherein the pre-processing coefficients are curved substantially according to a function $1/(2 - \cos(t))$.

7. (original) The method of claim 4 wherein the pre-processing coefficients form a pure window function.

8. (original) The method of claim 4 wherein the pre-processing coefficients have a width that is substantially coincident with the width of each of the input data blocks and an interval between one of the input data blocks and its ghost.

9. (original) The method of claim 4 wherein step a) comprises the step of down sampling each of the input data blocks, wherein step c) comprises the step of down sampling each of the input data blocks, and wherein step d) comprises the step of down sampling each of the input data blocks.

10. (original) The method of claim 9 wherein the pre-processing coefficients are curved.

11. (original) The method of claim 9 wherein the pre-processing coefficients are curved substantially according to a function $1/(2 - \cos(t))$.

12. (original) The method of claim 9 wherein the pre-processing coefficients form a pure window function.

13. (original) The method of claim 9 wherein the pre-processing coefficients have a width that is substantially coincident with the width of each of the input data blocks and an interval between one of the input data blocks and its ghost.

14. (original) The method of claim 4 further comprising the step of h) applying a spectral transformation between the step g) and the steps a), c), and d), wherein the spectral transformation is longer than an input data block.

15. (currently amended) A method of substantially eliminating a ghost of a received main signal and reducing noise enhancement comprising the following steps:

a) processing the received main signal and the ghost along n paths to produce n processed main signals and n processed ghosts, wherein each of the n paths

includes a corresponding finite filter, wherein the
processing along each of the n paths does not
substantially eliminate the ghost, wherein $n > 3$, and
wherein the processing along at least some of the n paths
includes shifting data; and,

b) adding the n processed main signals and the
 n processed ghosts such that, because of the addition of
the n processed main signals and the n processed ghosts,
the ghost of the received main signal is substantially
eliminated.

16. (original) The method of claim 15 wherein
step a) comprises the step of down sampling data in at
least some of the n paths.

17. (original) The method of claim 15 further
comprising the step of applying a spectral transformation
prior to step a).

18. (original) The method of claim 15 further
comprising the step of c) applying pre-processing
coefficients to the received main signal and the ghost
prior to step a).

19. (original) The method of claim 18 wherein the pre-processing coefficients are curved.

20. (original) The method of claim 19 wherein the pre-processing coefficients have a width that is substantially coincident with the width of the received main signal and an interval between the received main signal and the ghost.

21. (original) The method of claim 18 wherein the pre-processing coefficients are curved substantially according to a function $1/(2 - \cos(t))$.

22. (original) The method of claim 18 wherein the pre-processing coefficients form a pure window function.

23. (original) The method of claim 18 wherein the pre-processing coefficients have a width that is substantially coincident with the width of the received main signal and an interval between the received main signal and the ghost.

24. (original) The method of claim 18 wherein step a) comprises the step of down sampling data in at least some of the n paths.

25. (original) The method of claim 18 further comprising the step of applying a spectral transformation between step a) and step c).

26. (currently amended) An equalizer for processing blocks of data comprising:

n processing paths arranged to process the blocks of data;

$n - 1$ data shifters, wherein each of the $n - 1$ data shifters is in a corresponding one of the n processing paths so that one of the n processing paths has no data shifter;

n finite filters, wherein each of the n finite filters is in a corresponding one of the n processing paths, wherein each of the n finite filters applies a corresponding set of finite filter coefficients to the blocks of data, wherein ghosts of the blocks of data are not eliminated as a result of the application of the sets of finite filter coefficients corresponding to the n finite filters, and wherein $n > 2$; and

an adder arranged to add outputs from the n processing paths, wherein the addition eliminates ghosts of the blocks of data.

27. (original) The equalizer of claim 26 further comprising a spectral transformation applied upstream of the n processing paths, wherein the spectral transformation is longer than a block of data.

28. (original) The equalizer of claim 26 further comprising a pre-processor that applies pre-processor coefficients to each data block upstream of the n processing paths.

29. (original) The equalizer of claim 28 wherein the pre-processing coefficients have a width that is substantially coincident with the width of one of the data blocks and an interval between the one data block and its ghost.

30. (original) The equalizer of claim 28 wherein the pre-processing coefficients are curved.

31. (original) The equalizer of claim 30 wherein the pre-processing coefficients have a width that is substantially coincident with the width of one of the data blocks and an interval between the one data block and its ghost.

32. (original) The equalizer of claim 28 wherein the pre-processing coefficients are curved substantially according to a function $1/(2 - \cos(t))$.

33. (original) The equalizer of claim 28 wherein the pre-processing coefficients form a pure window function.

34. (original) The equalizer of claim 28 further comprising a spectral transformation applied downstream of the pre-processor and upstream of the n processing paths, wherein the spectral transformation is longer than a block of data.

35. (original) The equalizer of claim 26 further comprising n down samplers, wherein each of the n down samplers is in a corresponding one of the n processing paths, and wherein each of the n down samplers down samples each data block.

36. (original) The equalizer of claim 26 wherein half of the $n - 1$ data shifters shifts data in the data blocks to the left, and wherein half of the $n - 1$ data shifters shifts the data in the data blocks to the right.

37. (original) The equalizer of claim 26 wherein n is an odd integer, wherein half of the $n - 1$ data shifters shifts data in the data blocks to the left, and wherein half of the $n - 1$ data shifters shifts the data in the data blocks to the right.

38. (original) The equalizer of claim 26 wherein $n > 4$.